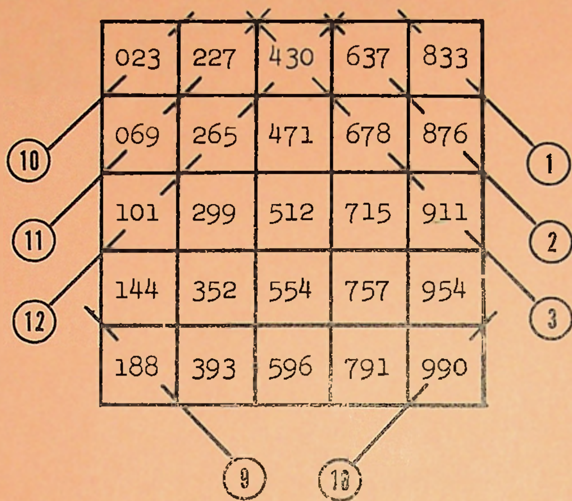
31



Average Variances — A Contest

The 5 x 5 array shown here contains 25 numbers in the range between 023 and 990. There are nine diagonals in the positive direction in the array (four of them are numbered 1, 2, 3, and 9) and nine diagonals in the negative direction (four of them are numbered 10, 11, 12, and 18). For each of the 18 groupings of the cells in the array, the variance of the numbers is calculated (actually, only 14 calculations need to be made, inasmuch as the groupings 1, 9, 10, and 18 have a variance of zero).

(Problem continued on page 2)

OCTOBER 1975 VOLUME 3 NUMBER 10

Popular Computing

For example, for the numbers in group 3, the calculation is as follows:

430	184900
678	459684
911	829921
2019, sum	1474505, sum of squares

N = 3

and, using the formula for variance:

$$V = \frac{N\,\Sigma X^2 - (\Sigma X)^2}{N^2}$$

$$V = \frac{3(1474505) - (2019)^2}{9}$$

$$= 38572.666$$

The variances for the nine positive groupings are calculated and averaged. For the array as shown, this average is 53387.97 (taking one-seventh of the sum of the variances; that is, not counting paths 1 and 9). Similarly, the variances for the negative groupings are calculated and averaged; this average is 23740.84.

The difference between those two averages is 29647.13. That difference can be made much larger by rearranging the 25 elements of the array.

For the arrangement that produces the largest difference between the two average variances, a prize of $25 is offered. Up to two prizes may be awarded in the case of duplicate winning entries. Entries must be received by December 31, 1975. Your entry should include the following:

1. The difference between the two average variances.
2. The 5 x 5 array arrangement.
3. The sums and sums of squares of the 14 groupings, numbered as shown on the cover drawing.
4. The 14 individual variances, in two groups of 7. □

PROBLEM 106

POPULAR COMPUTING is published monthly at Box 272, Calabasas, California 91302. Subscription rate in the United States is $18 per year, or $15 if remittance accompanies the order. For Canada and Mexico, add $4 per year to the above rates. For all other countries, add $6 per year to the above rates. Back issues $2 each.

EXACMATH

The California State University and Colleges computing centers make available to users of the Control Data equipment a packaged program for high precision arithmetic called EXACMATH. The package was devised by Lee Armer and Robert Shafron in 1971. It is written in the COMPASS assembly language, and operates interpretively. Modifications to include dynamic storage allocation were made by David Babcock.

The user of EXACMATH specifies his desired level of precision, from 1 to 2000 decimal digits. All operations are in scientific notation, with an exponent range on 10 from -9999 to +9999. The package contains a pseudo-accumulator (PAC), a pseudo-MQ (PMQ), and an index register. The following op-codes are available (the mnemonics are the same as those in COMPASS):

LDA	Load PAC
STA	Store PAC
LDQ	Load PMQ
STQ	Store PMQ
ENA	Load (immediate) PAC
ADA	Add to PAC
SBA	Subtract from PAC
MUA	Multiply
DVA	Divide
INA	Add (immediate)
AZJ,LT	Jump on minus
UJP	Unconditional jump
RTJ	Link to subroutine
ENI	Load (immediate) index register
IJI	(immediate) increment index register
JMP,ZRØ	Test index register for zero
INI	Print 50 digits per line
XØA	Exit from EXACMATH
HLT	Halt

The original version of EXACMATH, using one word of the CDC 3170 per decimal digit, executed EXACMATH programs rather slowly. For example, a 2000-digit addition to the PAC took .5 seconds, and a 2000 x 2000 digit multiplication took 87 seconds. The current version executes somewhat faster.

Report on Current Equipment

Attention should be called to the general-purpose, low-speed, high-capacity computing machines, long mass-produced in this country and abroad by Jehovah Instruments.

The machine is normally furnished with not less than five input devices which can accept not only numeric and alphabetic information but a wide variety of other types of data, some of which is not relevant to the problem under attack. Output can be oral or written, or can be a true decision-making function. One form of decision-making is unique to this machine: the ability to decide its own start and stop times, as well as the choice of problem to be processed at any given time.

The main (and only) memory is the outstanding feature of the computer, with random access of some hundreds of millions of bits, housed in a small box (less than one cubic foot) at one end of the machine; total power dissipation is less than two watts. No special cooling system is required, and the machine can function efficiently over a wide range of temperatures.

Access time to the memory is rather slow, on the order of several hundred milliseconds; at times the access is more random than is desirable. This characteristic is under investigation, with some hope of improvement.

The arithmetic unit is apallingly slow, and is extremely limited in its range. Most models are restricted to two-digit arithmetic (with sign), although some units have gone as high as 12-digit arithmetic plus direct calculation of some elementary functions, mainly powers and roots. Rudimentary subroutining is automatic, and a stringently limited form of floating point operation is possible. Programming is always done in a very high level (VHL) language, part of which varies from country to country. The syntax of this language is not completely worked out and has hundreds of known bugs. Much research into its pathology goes on, but useful results seem to be far off.

Another unique characteristic is that each unit tends to improve with age; in fact, the 1975 models, just now being delivered, are nearly useless. Currently, the 1953 models are just beginning to produce, and the 1942 models are coming up to the peak of their efficiency. Models prior to those of 1890 are not recommended for extended use.

No provision has been made for self-checking circuitry, although the machine is outstanding in its discriminating circuits, and can apply tests of reasonableness to input data to a degree not attained in competitive equipment. A singular weakness is the tendency to invert digits in the read head. Models with two read heads are sought by museums.

Construction is unusually rugged and reliable--many models have been in steady operation for over 80 years--although nearly all components suffer from fatigue effects and must undergo periodic rest periods to recover; during these rest periods, portions of the machine continue to operate unattended. Most units exhibit an inverse read-around ratio, in that too great a time lapse between successive references to the same portion of memory leads to unwanted random digits. These are not generated by one of the standard generating routines, but by a method as yet unpublished.

Both the central processing unit and the peripheral units are unconditionally guaranteed for the life of the machine; if accidentally damaged externally, however, no replacement is presently available.

Rental of the machines (they cannot be purchased) varies over a wide range, from nearly zero to upwards of several hundred thousand dollars per year, depending on supply and demand and other factors. Normally one can expect a minimum of 2000 hours per year of good time, although some units more than double that figure. Generally, management is willing to pay the higher rentals for the units which produce more per year, but not in direct proportion. Unscheduled down time is extremely low, since the machines schedule most of their own down time.

Delivery on future units is promised in about nine months; no significant change in subsequent models is planned. The production rate of the two main types is currently quite high, and may, in fact, be increasing.

Reprinted, with minor changes, from
Computing News 70, February 1, 1956.

PC31-6

Wallis' Equation

2.094551481542326591482386540579302963857306105628239180304128
529045312189983483667146267281777157757860839521189062963459
845140398420812823701739655313940554761602258281889491443972
226659155954503993324264722568009386991773958828940161818349
274541496089412255766427423189176266353756703834005969977982
699322413195065772518831774755040777030564533960069893890400
070810745419565650796790773428970370611769913813663645137969
148005146490994859886316244171338183676927256810935234393350
829021121155782138749345906572369942850322802805468370491984
150943751699368228880404238776144669949690861886671171462763
831193084122200029962226913191413331347219645497049865725763
447045813361595564314263515666435990612814931236366715460209
242064599742472215407633500611525831039490163885590767859581
355422712975046921435180823705307943804740765059120475880257
028659458723292283017255980177394131472173006682363401718987
927286062606462579444564700707765876362014287541017356493027
412316449397202871680048920020175337494195218857458143191263
785335029104192302267995758972484277181123317126496134259372
803150072055724528647205697358537123564471188375586781178566
975832650224702103685024982453113673800809807839215172995065
662172862135761239152541671424944916444987617122983326575014
903558116683044811815400299594905078370245705472134217246580
272097750389014858643990003158121069157987982157367591836132
007087431791457342410241369117217675009610495638226692884756
441789690696276213867694704534524549752178657932387682520808
562708102993293308924226016658861673755229300210645516315229
025069802191939364053220740803817366558276650290702423615472
711141265066146166850587761118061311652153845655581129964828
336960942836188964923922007240191473105794862943351625841217
869995738802564922040654383594614224677806881334384613161865
158401451864165749169724552339107323780612415091457735133301
796112652668572530191816807174537676481469914465483710882012
303717097752539121123794894909505458066633594745555353425245
7221960571131256767

An example of high precision arithmetic. The above is the real root of Wallis' equation

$$x^3 - 2x - 5 = 0$$

correct to 2000 significant digits. The calculation was made by Stephen Marcus in 1971.

Whittaker and Robinson (1924) reported "A pupil of DeMorgan by Horner's method found the root of Wallis' well known example...to 51 places... This was subsequently extended to 101 decimal places; cf *The Mathematician*, 3 (1850), p. 290.

Log 31	1.4913616938342726796667041001184157223037015583041 85
Ln 31	3.4339872044851462459291643245423572104499389304805 92
$\sqrt{31}$	5.5677643628300219221194712989185495204763933775704 14
$\sqrt[3]{31}$	3.1413806523913930044930758964627499263508597185007 26
$\sqrt[10]{31}$	1.4097307383555407027096739385717256489039883077335 20
$\sqrt[100]{31}$	1.0349362928747379641459479851213739991989362793836 77
e^{31}	29048849665247.425231085682111679825666764695090296980249340835647455635037288098511817 6096
π^{31}	2580156526864958.5104040373422537002781424395644105406554761858397131034250416440883331 98
$\tan^{-1} 31$	1.5385494443596426991404002160889046026493862563647 29

Strategy for NOTONE

The game of NOTONE was described in PC29-13:

Two players take turns tossing two dice. When it is a player's turn, he tosses the dice and establishes his point. He may then toss the dice as many times as he wishes, and his score for the turn is the sum of the tosses. If his point reappears, however, his turn ends with a score of zero. What is the proper strategy of play for this game?

The following analysis is by Richard Hamming. The strategy must tell you what to do each time you face a toss of the dice and have already an accumulated amount A and have a given point P. Do you toss and risk either a gain or a total loss of A, or do you quit and keep A? Thus, the state of the game is a pair of numbers P and A, and the strategy gives "toss or quit." We calculate the expectation of the next toss.

If the next toss is not P, you will gain

$$\sum_{k \neq j} kP_k = \sum_{k=2}^{12} kP_k - jP_j = S - jP_j$$

On the other hand, if the point occurs you lose the amount A, and this occurs with probability P_j.

The total expectation is thus the difference, expected gain minus expected loss:

$$(S - jP_j) - AP_j$$

The calculations are given in the accompanying table. The strategy is thus: toss if the expectation (the column headed equivalent in the table) is positive, and take your winnings if it is negative. If equal, quit and save time, or toss and amuse yourself; it is a matter of indifference.

Point k	2	3	4	5	6	7	8	9	10	11	12
Probability of k:	1/36	2/36	3/36	4/36	5/36	6/36	5/36	4/36	3/36	2/36	1/36
$k \cdot p_k$	1/18	3/18	6/18	10/18	15/18	21/18	20/18	18/18	15/18	11/18	6/18

The sum, S, of the line above is 126/18

Expectation of gain	125/18	123/18	120/18	116/18	111/18	105/18	106/18	108/18	111/18	115/18	120/18
Expectation of loss	A/36	2A/36	3A/36	4A/36	5A/36	6A/36	5A/36	4A/36	3A/36	2A/36	A/36

Point P	36(expectation) = 36(difference)	Equivalent
2	250 - A	250 - A
3	246 - 2A	123 - A
4	240 - 3A	80 - A
5	232 - 4A	58 - A
6	222 - 5A	44.4 - A
7	210 - 6A	35 - A
8	212 - 5A	42.4 - A
9	216 - 4A	54 - A
10	222 - 3A	74 - A
11	230 - 2A	115 - A
12	120 - A	120 - A

Capitalizing on Square Root

Given the capability to take square roots (which is the most common function on pocket calculators after the arithmetic functions), other roots can be calculated readily, using the number itself as the starting value and repeated square roots and multiplications. To extract cube roots, for example, of a number X:

Take two successive square roots of X
Multiply by X
Square root twice
Multiply by X
Square root twice
...and so on to convergence

2.359611
73.147941
2.9244925
90.659267
3.0856953
95.656554
3.1273657
96.948336
3.137871
97.274001
3.1405028
97.355586
3.1411611
97.375994
3.1413257
97.381096
3.1413668
97.382370
3.1413771
97.382690
3.1413897
97.382770
3.1413803
97.382789
3.1413805
97.382795
3.1413806
true value: 3.141380652..

For the cube root of 31, for example, the sequence shown here develops by this process. The process can be expressed this way:

$(X)^{1/3}$ = SSXSSXSSXSSXSSXSSXSSX ←

where each S stands for a square root and each X stands for a multiplication; the operations proceed from right to left. The sequence for any root can be derived by expressing the exponent in binary:

1/3 = .01010101010101010101... binary

and replacing each 1 bit with 01:

001001001001001001001001...

and then letting each zero stand for S and each 1 for X. For example, for seventh root,

1/7 = .001001001001001001001... binary

So for seventh root, use the sequence

...SSSXSSSXSSSXSSSXSSSXSSSX

For the 100th root,

$.01)_{10} = .000000101000111101011100001010001111\ldots$ binary

which leads to the sequence of operations:

SSSSSSSXSSXSSSSXSXSXSSXSSXSXSXSSSSSXSSXSSSSSXSXSXSX

and the sequence for 10th root:

SSSSXSXSSSXSXSSSXSXSSSXSXSSSXSXSSSXSX

Herman P. Robinson notes, "In each case, after the error is small, each square root reduces the error by a factor of 2. Thus, in taking the cube root, the error is reduced by a factor of 4 each iteration."

Mr. Robinson adds, "(The algorithm) is much more general than you think, because it works for any base and, in fact, can be useful when the exponent is not rational.

An example using the exponent 1/7 and working to base 10 gives an insight into how the method works:

$$1/7 = .142857142857\ldots$$

We have $y = Tx^1Tx^4Tx^2Tx^8Tx^5Tx^7Tx\ldots$ If any zeros had occurred in the fraction, they would be replaced by Tx^0. T stands for the operation of exponentiation to the 1/(base) power. For binary, T is the square root. In the present case, T is the 10th root. Since the above sequence is infinite, the part after the first cycle is the same as the entire part, so

$$y = TxTx^4Tx^2Tx^8Tx^5Tx^7y.$$

Raising to powers of 10, we get

$$y^{10} = xTx^4Tx^2Tx^8Tx^5Tx^7y$$

$$y^{100} = x^{10}\cdot x^4Tx^2Tx^8Tx^5Tx^7y$$

$$\vdots$$

$$y^{10^6} = x^{((((10+4)\cdot 10+2)\cdot 10+8)\cdot 10+5)\cdot 10+7}\cdot y$$

$$= x^{142857}y$$

or $$y^{999999} = x^{142857}, \quad \text{whence } y = x^{1/7}$$

Note that the denominator of the exponent, 999999, is one less than the Nth power of the base, in this case 10, where N is the period length." □

Book Review

The Thorndyke Encyclopedia of Banking and Financial Tables

by David Thorndyke, Warren, Gorham & Lamont,

Boston, 1973, $47.50.

The art of table making is now almost lost, since it is feasible and efficient to compute whatever it is you want when you want it. And if what you want happens to be compound interest calculations, there are pocket calculators that give you any one of the four basic compound interest formulas at the touch of a button.

Nevertheless, there is still a place for financial tables, since they provide data in a form that allows for comparisons and "mathematical browsing." To be sure, nearly every entry in the Thorndyke Encyclopedia could be calculated readily and cheaply, but having them all displayed in orderly fashion provides a way to make comparisons (say, between alternative interest rates) that would be tedious with independent calculations.

The 1388 (8.5 x 11) pages form 24 tables, which include:

Compound interest and annuity tables.
Depreciation schedules.
Mortgage amortization schedules.
Days between dates.
Growth of 1.
Savings growth.
Annual percentage rates.
Installment loan payments.
Discount values.
Stock yield.

The scope of rates, yields, and terms is tremendous--both coupon and yield rates range from 0% to 12%; interest rates from 1 to 100%; terms extend to 40 years. All tables are computer-generated and have been thoroughly checked for complete accuracy. Each table includes a step-by-step example illustrating how to use that specific table.

There are more tables than any one person is likely to use, so the book could serve several different people in one office. One person, alone, would probably use some tables frequently and ignore others. In order to use the tables effectively, one might have to spend a little time getting acquainted with the arrangement.

This is not a book for a CPA to carry in his briefcase to a client's office, but it is an excellent reference volume to have in his own office. Similarly, it might not be suitable for a bank's loan officer to use daily in conference with borrowers, but would be good for a borrower to use before going to the bank. □

A Random Number Generator

The flowchart (Q) shows the scheme for a simple random number generator subroutine to use on the Altair 8800. The basic scheme is shown in flowchart (W), in which a starting value, X, is doubled and reduced by a modulus. There are five such basic generators, and their output is summed modulo 100 into a word S. The five moduli are 13, 19, 23, 53, and 59, so that each basic generator has a full size cycle length, except for 23, which has a cycle length of 11. For example, for p = 13, the basic generator produces as output the sequence

2, 4, 8, 3, 6, 12, 11, 9, 5, 10, 7, 1

endlessly, for a cycle length of 12. The cycle length for the five generators beaten together is the least common multiple (LCM) of their individual cycle lengths (in this case, 149292). The cycle length for the complete subroutine is found by the following analysis (due to David Ferguson).

The moduli, p_i	13	19	23	53	59
Individual cycle lengths, c_i	12	18	11	52	58
Least common multiple of the c_is:			(149292)		
Sum contributed to S by one full cycle of each generator, t_i (mod 100)	78	71	92	78	11

$t_i = (Y + 1) \cdot p_i$ (mod 100), where Y is the number of times that the path of flowchart W passes reference 2 for that generator

Number of cycles of each generator before all generators get back to 1, r_i	41	94	72	71	74

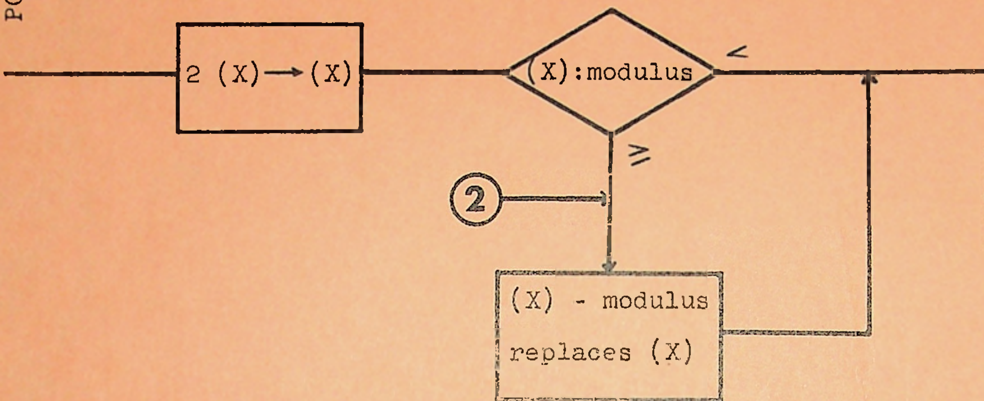
$r_i = (\text{LCM})/c_i$ (mod 100)

$r_i \cdot t_i$ (mod 100)	98	74	24	38	14

The sum of $r_i t_i$ (mod 100) is 48 = B

The total cycle length, then, is $\frac{100}{(B,100)} \cdot \text{LCM} = 3732300$

where this is the gcd $\longrightarrow$ (B,100)



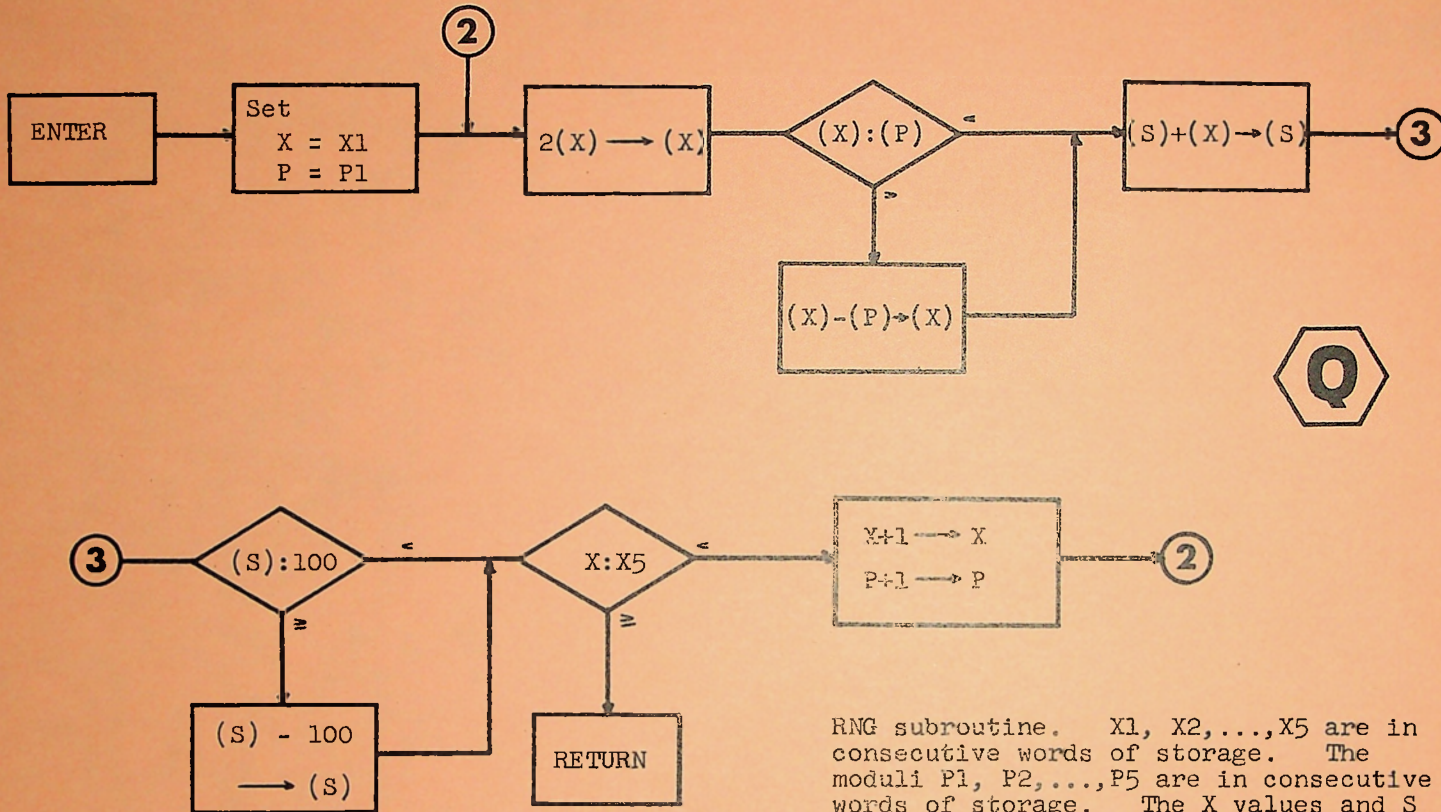
In similar fashion, the cycle length for the subroutine when the p values are 3, 7, 11, 13, and 17 is as follows:

p_i	3	7	11	13	17
c_i	2	3	10	12	8
LCM			(120)		
t_i	3	7	55	78	68
r_i	60	40	12	10	15
$r_i t_i$	80	80	60	80	20

The sum of $r_i t_i$ is 20

Total cycle length:

$$\frac{100}{(20,100)} \cdot 120 = 600$$



RNG subroutine. X1, X2,...,X5 are in consecutive words of storage. The moduli P1, P2,...,P5 are in consecutive words of storage. The X values and S are initialized in the main routine. Output is a random number between 00 and 99 in word S. For the P values 13, 19, 23, 53, and 59, the cycle length is 3732300 numbers. Parentheses indicate (contents at address). □

Book Review

COMPUTER PROGRAMMING: TECHNIQUES, ANALYSIS, AND MATHEMATICS

by Richard V. Andree, Josephine P. Andree, and David D. Andree.
Prentice-Hall, 1973, xvii + 549 pages, $12.95.

Reviewed by Edward A. Ryan, Woodland Hills, California.

Covering as much material as indicated by the title would be an ambitious project for any book, let alone one that may be a student's first brush with the world of computing. It would be to the authors' credit to say that they had accomplished their objective, but such praise is not deserving for this book. The book indeed covers all three areas mentioned in the title. However, it is heavily oriented toward programming techniques, touches only lightly on problem analysis, and mentions mathematics only enough to justify the title.

The Andrees state in the preface to the instructor that "This book was meant to be read by students." I cannot argue with this statement. It is obvious that the book represents a compendium of lecture notes and classroom discussions collected over a long period of time. Now, there is nothing inherently wrong with this idea. What makes it wrong for this book is that there was not sufficient editing between the professor's notebook and the published copy. The text is replete with typographical errors and other inaccuracies. I stopped keeping track somewhere in the third chapter.

Any book which is intended to teach students "efficient programming techniques" should not also teach them bad habits. There are two outstanding areas in which the book deserves criticism for doing just that. The first is in the flowcharting style and symbols used throughout the text. The authors have chosen to ignore the fact that an ANSI standard for flowcharting exists. Instead, they have selected a notation that a friend who looked at the book described as "just awful." In all fairness, the flowcharts are very readable and I had no trouble understanding any of them. The point is that their style is antiquated to say the least, and has no place in a computing text of the 70's.

The other area of the book which teaches bad habits is the sample programs used to illustrate coding techniques. This is more serious, since coding habits will tend to become more ingrained in a person's computing personality than flowcharting techniques. A better picture of a problem will have more effect than a listing of a better program. Many of the programs have no comments or other internal documentation. In many of the reproduced program listings, the statement numbers are embedded with the C's in comment statements, making program branches difficult to follow. There is not the slightest hint of style in any of the myriad of sample programs used throughout the text. I submit again that if we are going to teach kids the rudiments of programming, let's teach 'em right.

Lastly, I think this book deserves criticism for its treatment of assembly language programming. This chapter appears to have been included almost as an afterthought. It bears no relation to the rest of the book and, after only the briefest of introductions to the IBM 1130, it goes into some very sophisticated examples that would test the ability of many an experienced practitioner to understand. The rationale for this chapter is questionable.

Now that I have thoroughly chastised the book and its authors, let me inject several notes of well deserved praise. I managed to learn a great deal about computing--practical computing--from reading this book. Contained within its 500-odd pages is a wealth of information and experience that can come only from long hours of sweating over a hot console. It shows in every chapter. The set of practice problems is outstanding and makes the book worth its value at twice the price. There are two chapters (one on simulation, the other on numerical methods) which treat their respective subjects with a depth and sensitivity not ordinarily encountered in an introductory text.

As a final note let me say that this book has its contribution to make to the literature of the field, although I believe it to be of more value to the more experienced user than to the beginner.

Editor's note: Many of the errors that Mr. Ryan noted have been corrected in the book's second printing. Also, an instructor's manual is available, which discusses the philosophy of what was done in the text, and why. □

Problem Solution

Problem 32, Book Page Numbering, appeared in issue No. 10:

Book pages are consecutively numbered from one to K. Using individual pieces of type, how many of each of the decimal digits will be used to number the pages?

Raymond Clare, Indianapolis, performed the counts for values of K up to 50,000, using a Monroe 1880. Results for various values of K are given here:

K	0	1	2	3	4	5	6	7	8	9
100	11	21	20	20	20	20	20	20	20	20
147	24	83	35	35	33	25	25	25	24	24
1000	192	301	300	300	300	300	300	300	300	300
10000	2893	4001	4000	4000	4000	4000	4000	4000	4000	4000
25000	9392	20500	15501	10500	10500	9501	9500	9500	9500	9500
30000	10893	22000	22000	12001	12000	12000	12000	12000	12000	12000
50000	18893	30000	30000	30000	30000	20001	20000	20000	20000	20000